Related Work is All you Need

Rodolfo Zevallos[*1], **John E. Ortega**[*2], **Benjamin Irving**[*2]
Universitat Pompeu Fabra[*1], Northeastern University[*2]
Barcelona-Spain, Boston-USA
rodolfojoel.zevallos@upf.edu, {j.ortega, irving.b}@northeastern.edu

Abstract

In modern times, generational artificial intelligence is used in several industries and by many people. One use case that can be considered important but somewhat redundant is the act of searching for related work and other references to cite. As an avenue to better ascertain the value of citations and their corresponding locations, we focus on the common "related work" section as a focus of experimentation with the overall objective to generate the section. In this article, we present a corpus with 400k annotations of that distinguish related work from the rest of the references. Additionally, we show that for the papers in our experiments, the related work section represents the paper just as good, and in many cases, better than the rest of the references. We show that this is the case for more than 74% of the articles when using cosine similarity to measure the distance between two common graph neural network algorithms: Prone and Specter.

Keywords: GNN model, Related Work, knowledge graphs

1. Introduction

Authors and researchers often turn to tools like Google Scholar[1], Semantic Scholar[2], and ResearchGate[3] to find related research for their scientific papers. These tools rely on knowledge graphs, where each paper is a node representing common sections, with the "related work" (RW) section being crucial for computational scientific articles.

In order to best represent a research article Q an author would typically have to use search tools to find references to create a RW section for Q. The RW section would contain text and cite other articles similar to Q. In order to automate such a process, we propose as a first step a novel technique based on a newly-released corpus that contains human annotations of related work sections. We demonstrate that when using two common graph neural network (GNN) algorithms to encode research articles, (1) Prone (Zhang et al.) and (2) Specter (Cohan et al., 2020), the RW section of Q is more similar to a pre-trained GNN embedding than other sections, hence "related work is all you need" for representing Q in a GNN. In this article, we denote the encoded GNN vectors of an outgoing citation (called FANOUT) from Q as (V_{FANOUT}). The representation of an article in a GNN consists of citations as graph edges and articles as nodes. We show that the related work section of an unseen article in most cases is more similar to its pre-trained encoding than other sections of the article. This is achieved by comparing V_{FANOUT} to the original Prone or Specter GNN embeddings (Q_{model}) (held out pre-trained models).

To better summarize our approach, we first review previous work in Section 2, followed by an explanation of how citation graphs are employed in tools like Semantic Scholar and others. We then detail our methodology using GNNs in Section 3 and provide experimental settings in Section 4. Finally, Section 5 presents the results, and Section 6 discusses the other models, concluding that "related work is all you need."

2. Related Work

The idea that related work could be the best FANOUT representation of a paper has not been explored heavily if at all. However, there has been a considerable amount of work in related work *generation* that attempts to create the FANOUT of the related work section of a paper.

For example, Chen et al. (2021) uses a set of publications that are related to Q and summarizes them to give the author more insight into other works. Their work extracts sentences from papers and generates an abstract from the related publications. Their work uses a relation-aware effect based on relations from a graph with a unique dataset. Unfortunately, since their graph is unique and not part of a major benchmark, it differs from our approach which uses graphs that are commonly used by Prone and Specter.

On the other hand, a technique similar to Prone and Specter uses text and citations to generate related work. (Chen and Zhuge, 2019) Their work is more closely aligned to ours because, while it does not show that related work is more important, it attempts to generate the related work from the text

[1] https://scholar.google.com/
[2] https://www.semanticscholar.org/
[3] https://www.researchgate.net

LREC-COLING 2024, pages 13874–13878

of a paper to other papers that cite Q. Keywords are extracted from those papers called Q_{FANIN} and constructs an intermediate graph that is used to generate a related work section. Their work uses ROUGE, MEAD, and LexRank for evaluation of the generated related work. In our work, citations and text are compared but the similarity metric used is based on cosine similarity. Our goal is to show that the RW section is a more powerful representation, thus, fluency metrics and others for generation are not used for evaluation.

Somewhat related is the work from Li and Ouyang (2022) that surveys the field according to more recent summarization techniques. Comparisons are done on the latest approaches. Their work shows important to the related work section and shows that several approaches are based on tasks that are not directly related to what we propose. However, it seems as if several of the extractive techniques have found citations helpful while not using citation graphs as a method of training. Our work explicitly uses the information gotten from citation graphs that were created using the Prone and Specter method.

One work (Lv et al., 2022) is most closely related to our work because it uses a a large graph of papers along with a learning method that projects citations and papers into a vector space, thus making them measurable via a linear vector-space method such as cosine similarity. Their work creates a graph that is derived from titles and abstracts, similar to the Specter method compared here. However, their work classifies entire papers into scientific categories omitting a section-level analysis.

Mai et al. also conducted experimentation on datasets from Semantic Scholar (similar to the Specter method) to validate entity retrieval where entities are mostly paragraph vectors. Additionally, they note that knowledge graph embeddings along with paragraph vectors can be useful for link prediction. Their work touches on several of the same themes as ours but does not explicitly show that paragraphs belong to specific sections nor do the make claims of the value of FANOUT from specific sections.

Another approach (Jia and Saule, 2018) creates clusters citations based on distributed representations of papers. Instead of doing random walks like Prone they construct neighborhoods using a sliding window over consecutive words in a sampling procedure. Their approach is impressive and should be considered a good next step to show the value of the citations chosen. In this work, we focus strictly on showing that for two baseline graph embedding approaches we are able to show that the combination of V_{FANOUT} in RW alone is a good representation of Q.

3. Methodology

3.1. Graphs Embeddings

Language models trained using algorithms like Prone and Specter generate embeddings that represent articles. The embeddings are numerical represenations of graphs and are able to capture semantic relationships and information measurable by graph metrics such as cosine similarity. When encoding an embedding for an article (Q) that is not in a set of articles created by a pre-trained GNN Prone or Specter model, it may be difficult to generate a new embedding for Q. This is due to fact that Q is not part of the original graph and, thus, the lacks relationship and other knowledge – Q is not an article in G, the pre-trained graph. In order to resolve this issue, we propose that an unseen article Q_{new} can be represented by the centroid, or weighted average of its V_{FANOUT} (the vector of all of its outgoing citations) as illustrated in In Figure 1.

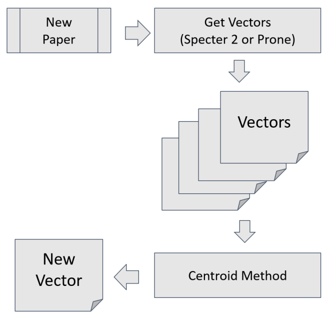

Figure 1: Flowchart illustrating the process of generating a new vector (Q_{new}) using the centroid method.

3.2. Related Work Representations

To demonstrate that the RW section annotated from the newly-formed dataset[4] accurately represents articles, we combine the V_{FANOUT} of related work sections, calculate their weighted average (centroid), and compare it with centroids from all sections using cosine similarity.

We employ two cutting-edge models for generating semantic vectors of articles: Specter and Prone. Specter, a transformer-based model (Vaswani et al., 2017), utilizes SciBERT (Beltagy et al., 2019) and PubMedBERT (Gu et al., 2020) language models, producing 1024-dimensional embedding vectors capturing semantic relationships among papers. Prone, based on the BERT architecture, is tailored for scientific documents, generating 512-

[4] https://github.com/rjzevallos/Related-Work-is-All-you-Need

dimensional embedding vectors representing text semantics.

For each article, we use the unique DOI id to obtain its Semantic Scholar ID (SSID)[5]. We use the SSID to query both pre-trained models (Specter and Prone) to obtain encoded GNN embeddings. Each paper's embedding is then used to compute the centroid of vectors from FANOUT related work sections and compare to other sections. A description follows in Algorithm 1.

Algorithm 1 Related Work vs. All Distance

.

Require: Q with N references: R_i

Require: Q_{model} generated for Prone and Specter Models

1: Generate V_{FANOUT} of Q using Prone and Specter models

2: Split V_{FANOUT} into V_{rw} (Related Work references FANOUT) and V_{all} (All references FANOUT)

Require: $V_{rw} = \{R_a, R_b, R_c\}$ where $R_{a,b,c} \in RW$

Require: $V_{all} = V_{FANOUT}$

3: Calculate the centroid vector of $V_{rw} \implies Q_{rw} = \frac{1}{N_{rw}} \sum_{i=1}^{N_{rw}} V_{rw_i}$

4: Calculate the centroid vector of $V_{all} \implies Q_{all} = \frac{1}{N_{all}} \sum_{i=1}^{N_{all}} V_{all_i}$

5: Calculate the cosine similarity between Q_{rw} and $Q_{model} \implies C_{rw} = \frac{Q_{rw} \cdot Q_{model}}{\|Q_{rw}\| \cdot \|Q_{model}\|}$

6: Calculate the cosine similarity between Q_{all} and $Q_{model} \implies C_{all} = \frac{Q_{all} \cdot Q_{model}}{\|Q_{all}\| \cdot \|Q_{model}\|}$

4. Experimental Setup

In this section, we detail the setup of the various experiments carried out in the context of our study on FANOUT representations in graph models for the analysis of articles. To evaluate the effectiveness of our approach, we conduct a series of experiments covering multiple aspects of the processing and application of these models.

4.1. Dataset

The dataset (made public)[6] used in this study consists of 400,000 academic research papers in the English language, spanning diverse fields of research including computer science, engineering, physics, chemistry, mathematics, economics, and others. The papers originate from peer-reviewed venues including journals and conference proceedings in various scientific disciplines. The corpus covers publications from the past 5 years as well as seminal historical papers within each field. Each paper contains the full list of bibliographic references, which have been manually categorized by expert annotators into related works versus other references. The related works are directly relevant to the paper's contributions and methodology, while other references provide background information.

On average, each paper has 20 references, with a minimum of 5 and maximum of 50 references per paper. 60% of the papers come from computer science conference proceedings including ACL, NeurIPS, ICML, CVPR, ICCV, SIGIR, WWW, and others. The remaining 40% come from scientific journals such as Nature, Science, IEEE Transactions on Pattern Analysis and Machine Intelligence, ACM Transactions on Information Systems, Journal of Machine Learning Research, and more. Each paper is uniquely identified by an ID mapped to its Semantic Scholar identifier, to retrieve pre-computed embedding representations from the Specter and Prone models.

4.2. Experiments

This section describes the experiments conducted to evaluate the hypothesis that the FANOUT from a RW section better captures semantic information of a Q_{new} when compared to other sections. We perform the following two main tasks: (1) assess the centroid of RW vectors from our dataset in the absence of modeling and (2) create a predictive model that will combine both the RW from papers and other sections to create vectors that can represent the entire paper.

4.2.1. Task One: Related Work Vectors

In order to better ascertain the validity of our hypothesis, we focus on first showing that the RW section represents an article well without modeling. We use the entire set of available articles (400K) by creating the V_{FANOUT} of each paper. This is done by computing the centroid of both the Specter and Prone vectors with the assumption that the centroid vector best summarizes what the RW section contains. We then measure the cosine similarity difference between: (1) Q_{rw} and Q_{model} and (2) Q_{all} and Q_{model} of each article. After that, we evaluate the results by calculating the average count of RW centroids (C_{rw}) and All centroids C_{all} that performed best as shown in Algorithm 1.

4.2.2. Task Two: Predictive Models

In our second task, we aim to assess how well a predictive model can estimate Q_{new} and compare it to the Q_{rw} assumption. We want to see if

[5]The original Specter GNN embeddings were obtained from Semantic Scholar (www.semanticscholar.com)

[6]https://github.com/rjzevallos/Related-Work-is-All-you-Need

combining Q_{rw} and Q_{nrw} still produces good results, similar to adversarial learning. We evaluate the models by measuring cosine similarity between predicted vectors ($Q_{predict}$) and Q_{rw} compared to the actual representation of each scientific paper. When the model predicts higher cosine similarity, we review the examples to find the best paper representation. This helps us determine if the models offer additional insights beyond related work alone. Our results provide insights into optimal V_{FANOUT} representation strategies.

For this task, we use two neural network approaches. The first is a feed-forward neural network (**NN1**), consisting of three hidden layers with 512 units and ReLU activations. It has 50% dropout and matches the embedding size (768). We use a learning rate of 0.0005 and a batch size of 32, with an Adam optimizer, and fine-tune hyperparameters via grid search.

The second approach, **NN2**, is also a feed-forward neural network operating on the same inputs. It comprises 100 hidden layers, each with 256 units and GELU activations. After each GELU activation, we apply layer normalization and 40% dropout. Input dimensions are 256 for the V_{rw} and 512 for all V_{FANOUT}. We use a learning rate of 0.00005, a batch size of 32, a CosineAnnealing learning rate scheduler, and an AdamW optimizer for training.

5. Results

In our experiments, we show that the creation of a RW vector (V_{rw}) is an effective approach for creating graph embeddings of an unseen article. The results in this section provide more detail into both tasks performed.

5.1. Task One: Related Work Vectors

In Task One, we calculate weighted average centroids of Prone and Specter embeddings for V_{rw} using only related work sections of each paper. We also compute separate centroids for Prone and Specter embeddings for V_{all}, considering all paper sections. Then, we compare the C_{rw} and C_{all} values to determine the final paper representation. “Related Work” FANOUT is more similar to the original encoding in 74% (Prone) and 69% (Specter) of all articles, respectively.

5.2. Task Two: Predictive Models

In Task Two, we leverage the prior knowledge gained in Task One as a valuable means to predict vectors that can enhance the representation of a scientific paper. Our objective in this task is to assess the feasibility of constructing a predictive model that combines both Q_{rw} and Q_{nrw} centroids to more accurately approximate the final Prone and Specter representations of the paper. To achieve this, we employ NN1 and NN2 as determining mechanisms.

Using NN1, a three-layer feedforward neural network, we observe a notable improvement when combining Q_{rw} with Q_{nrw} within the predictive model. Specifically, we achieve a 2.7% enhancement in the case of Prone and a 3.2% enhancement in the case of Specter, as measured by cosine similarity. Our evaluation, limited to the test set, validates that the predicted vectors better align with the paper's content compared to the vectors derived from related work in 62% of the cases. Notably, our experiments reaffirm the prominence of the "Related Work" section as the primary determinant for generating a GNN vector representation of a paper. However, it's important to underscore that a neural network can be employed synergistically with the "Related Work" section to enhance the outcomes of other FANOUT methods.

Conversely, NN2 yields even more impressive results, boasting an 8.1% improvement for Prone and a remarkable 10.15% improvement for Specter over the baseline Q_{rw}. Furthermore, our assessment demonstrates that, in 78% of evaluated cases, the predicted vectors outperform those derived from the related work section in terms of alignment with the paper's content.

6. Conclusion

In this article, we explore two major graph neural network methods: Prone (Zhang et al.) and Specter (Cohan et al., 2020). We demonstrate that for a newly-formed dataset where “related work” citations are annotated, the “related work” section alone more often than not (more than 50% of the time) represents scientific papers sufficiently. This insight provided a novel method for generating paper vectors.

Our experiments highlight the key role of related work citations (FANOUT) in shaping paper representations. Summarizing pertinent papers captures primary topical and semantic signals, reinforced by significant improvements in cosine similarity across both Prone and Specter embeddings.

We introduce predictive models that combine related work with other V_{FANOUT} to enhance GNN representations. Integrating related and non-related reference knowledge can surpass individual paper vectors' limitations, potentially creating superior composite representations. Our approach extends beyond encoding new papers, consolidating information into better encodings for unseen articles.

In conclusion, this work advances optimal reference-based strategies for academic document vector representations, guiding researchers in situating manuscripts within semantic vector spaces for exploration and discovery.

Acknowledgements

The first author has been supported by a FI grant of the Catalan Funding Agency for Research and Universities (AGAUR). Moreover, part of the work presented here was carried out at the 2023 Jelinek Memorial Summer Workshop on Speech and Language Technologies (JSALT) and was supported with funds from Johns Hopkins University and ESPERANTO, the EU Horizons 2020 program's Marie Sklodowska-Curie Grant No 101007666. (Amazon Alexa || Google Research || The Jean-Zay supercomputing center) provided some cloud computing support.

7. Bibliographical References

Zafar Ali, Guilin Qi, Pavlos Kefalas, Shah Khusro, Inayat Khan, and Khan Muhammad. 2022. Sprsmn: Scientific paper recommendation employing specter with memory network. *Scientometrics*, 127(11):6763–6785.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Jingqiang Chen and Hai Zhuge. 2019. Automatic generation of related work through summarizing citations. *Concurrency and Computation: Practice and Experience*, 31(3):e4261.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. 2021. Capturing relations between scientific papers: An abstractive model for related work section generation. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD: proceedings. International Conference on Knowledge Discovery & Data Mining*, volume 2016, page 855. NIH Public Access.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus).

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. 2021. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: an optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633.

Haofeng Jia and Erik Saule. 2018. Graph embedding for citation recommendation. *arXiv preprint arXiv:1812.03835*.

Xiangci Li and Jessica Ouyang. 2022. Automatic related work generation: A meta study. *arXiv preprint arXiv:2201.01880*.

Yiqin Lv, Zheng Xie, Xiaojing Zuo, and Yiping Song. 2022. A multi-view method of scientific paper classification via heterogeneous graph embeddings. *Scientometrics*, 127(8):4847–4872.

Gengchen Mai, Krzysztof Janowicz, and Bo Yan. Combining text embedding and knowledge graph embedding techniques for academic search engines.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. Prone: Fast and scalable network representation learning.